



UniCEUB – Centro Universitário de Brasília
FATECS – Faculdade de Tecnologia e Ciências
Sociais Aplicadas
Curso de Engenharia de Computação

Esteiras eletrônicas controladas por micro computador

**Aluno: Therence Barros de Oliveira
RA: 2007574/7**

Brasília - DF
Novembro de 2008

UniCEUB – Centro Universitário de Brasília
FATECS – Faculdade de Tecnologia e Ciências Sociais
Aplicadas
Curso de Engenharia de Computação

Esteiras eletrônicas controladas por micro computador

Aluno: Therence Barros de Oliveira
Orientador: Prof. Thiago Toribio

Monografia apresentada à Banca
Examinadora da Faculdade de Ciências
Exatas e Tecnologia UniCEUB, para
Conclusão do curso de Engenharia de
Computação.

Brasília - DF
Novembro de 2008

Dedicatória

Dedico este trabalho a meus pais, meus avôs, minha esposa e filha, pois sem eles eu não teria chegado até aqui.

Agradecimentos

Agradeço a todos os que passarão pela minha vida no período da minha graduação ajudando de alguma forma neste caminho.

Agradeço em especial a Jussara Vieira de Barros, minha mãe, pois só ela sabe o tamanho das pedras que me ajudou a remover neste trecho de caminho da minha vida, participou comigo de todas as dificuldades financeiras e psicológicas sempre a meu lado, e nos momentos mais difíceis em que pensei em desistir ela estava lá me ajudando e não deixando que a minha decepção e cansaço vencessem. Muito obrigado mamãe.

Agradeço também a minha esposa, que se juntou a mim no trecho final desta caminhada e suportou um homem estressado e cansado, e por amor entende estes percalços e ainda me presenteou com minha filha linda que passou a iluminar o meu caminho em um momento em que tudo parecia estar na escuridão. Obrigado amor.

Não posso esquecer também de agradecer a meus avôs que sempre foram um porto seguro e um exemplo em minha vida. Muito obrigado vovó e vovô.

Resumo

Este trabalho consiste na construção de um protótipo de uma esteira rolante, que funciona com a utilização de um motor de corrente contínua, que permanecerá em movimento enquanto objetos estiverem sobre a esteira. O controle do sistema é feito por meio de um computador. No momento em que um sensor de infravermelho detecta a entrada de um objeto na esteira rolante, o circuito ligado a este sensor envia um sinal ao computador através da porta paralela LPT1, um programa em Java controla a quantidade de objetos na esteira mantendo-a ligada ou desligada quando não é mais necessário. O circuito que intermédia a comunicação da esteira com o computador tem a capacidade de controlar sete esteiras sendo que neste projeto seis esteiras são simuladas através de uma maleta laboratório.

Palavras-chave: Porta paralela LPT1, sensor infravermelho, esteira rolante e motor de corrente continua.

Abstract

This work consists of the construction of an archetype of a rolling mat that functions with the use of a direct-current engine that remains in movement while objects are on the mat. The control of the system is made by means of a computer. At the moment where an sensory infra-red ray detects the input of an object in the rolling mat, on circuit to this pecker sends a sign to the computer through parallel port LPT1, a system in Java controls the amount of objects in the mat keeping this on one or disconnect when it is not more necessary. The circuit that intermediate the communication of the mat with the computer has the capacity to control seven mats being that in these design six mats will be simulated through one suitcase laboratory.

Key - Word: Parallel port LPT1, sensory infra-red ray, direct-current rolling and motor mat.

Sumário

LISTA DE SIGLAS	VII
LISTA DE FIGURAS	IX
LISTA DE TABELAS	X
CAPÍTULO 1. INTRODUÇÃO.....	11
1.1 <i>Motivação</i>	<i>12</i>
1.2 <i>Objetivo geral.</i>	<i>13</i>
1.3 <i>Objetivos específicos.....</i>	<i>14</i>
1.4 <i>Organização da monografia.....</i>	<i>14</i>
CAPÍTULO 2. DESENVOLVIMENTO DO PROTÓTIPO	16
2.1 <i>Tabela Verdade criada para o projeto.</i>	<i>17</i>
2.2 <i>Álgebra de Boole utilizada no projeto.....</i>	<i>19</i>
2.3 <i>Circuito Elétrico utilizada no projeto.....</i>	<i>21</i>
2.3.1 <i>Circuito elétrico dos sensores IR.</i>	<i>22</i>
2.3.2 <i>Projeto das portas lógicas utilizado no projeto.</i>	<i>24</i>
2.3.3 <i>Circuito elétrico que liga os motores das esteiras.</i>	<i>27</i>

2.4	<i>Circuito completo utilizado no projeto.</i>	29
2.5	<i>Ligação do circuito utilizado no projeto com a porta LPT1 do computador.</i>	29
2.6	<i>Programa Java que controla as esteiras.</i>	32

CAPÍTULO 3. CONSIDERAÇÕES FINAIS..... 36

3.1	<i>Resultados obtidos.</i>	36
3.2	<i>Dificuldades encontradas.</i>	38
3.3	<i>Conclusões.</i>	38
3.4	<i>Sugestões de trabalhos futuros.</i>	39
3.5	<i>Custo estimado do projeto.</i>	40

REFERÊNCIAS BIBLIOGRÁFICAS..... 41

Apêndice A – Esquema do Circuito Elétrico da Maquete	43
Apêndice B – Programa Desenvolvido na Linguagem Java	45
Apêndice C – Programa Desenvolvido na Linguagem Java	46
Apêndice D – Programa Desenvolvido na Linguagem Java	57

Lista de Siglas

XOR : Ou exclusivo;

LED: Diodo Emissor de Luz;

ANSI : Instituto Nacional Americano de Padronização;

IEC: Instituto Evandro Chagas;

COM: Porta de comunicação serial do computador;

USB: Porta de comunicação serial do computador;

IBM: Empresa Multinacional;

PC: Computador Pessoal;

LPT: Line Print Terminal;

SPP: Standard Parallel Port;

EPP: Enhanced Parallel Port;

ECP: Extended Capabilities Port;

CPU: Unidade Central de Processamento;

RAM: *Random Access Memory* (Memória de Acesso Aleatório);

BUS: Conjunto de vários fios paralelos responsáveis pelas conexões na CPU, na memória e nos demais dispositivos de entrada e saída de um computador. Um bus

envia dados nas duas direções, entre os dispositivos a que está conectado num sistema. Sua a velocidade de transmissão de dados é medida em MegaHertz (MHz).

JVM: (Maquina Virtual Java);

JNI : Java Native Interface (Interface Nativa Java);

SO: Sistema Operacional;

IR: Infravermelho;

CI: Circuito Integrado;

GND: Terra do circuito;

Lista de Figuras

Figura 1 Esteiras de transporte [16].....	11
Figura 2 Fluxograma de controle das esteiras.	12
Figura 3 Diagrama de bloco do projeto.	16
Figura 4 Maleta laboratório modelo “Minipa 500 em um”	17
Figura 5 Circuito montado para o projeto	22
Figura 6 esquema elétrico do circuito que gera o sinal dos 14 sensores IR	23
Figura 7 Circuito lógico do projeto	26
Figura 8 Circuito que liga os motores das esteiras	28
Figura 9 Esquema de funcionamento da entrada paralela do computador com o conector DB25 no modo SPP. [19]	30
Figura 10 Diagrama de blocos do programa <i>Java</i>	32
Figura 11 Tela do programa <i>Java</i>	33
Figura 12 Protótipo da esteira.....	36
Figura 13 Circuito de controle e maleta Minipa 500 em um.....	37

Lista de Tabelas

Tabela 1 Distribuição IR x Esteira.	18
Tabela 2 Tabela Verdade do projeto.....	19
Tabela 3 Endereçamento das Esteiras	20
Tabela 4 Tabela explicativa do circuito lógico.....	27
Tabela 5 Mapeamento Circuito Logico X Porta LPT1	31
Tabela 6 Mapeamento porta LPT1 X Esteiras.....	31
Tabela 7 Saída do programa <i>Java</i>	32
Tabela 8 Contador X Estado da esteira X Resposta do Programa.....	34
Tabela 9 Custo estimado do projeto	40

Capítulo 1. Introdução.



Figura 1 Esteiras de transporte [16]

Neste projeto é apresentado o desenvolvimento de um protótipo de automatização de uma esteira rolante acionada por um motor de corrente contínua que é controlado por um circuito interligado a um computador, o qual identifica a entrada e a saída de objetos da esteira, ligando e desligando o motor, gerando assim economia de energia.

O circuito de controle e o programa em *Java* desenvolvido têm a capacidade de controlar sete esteiras. O circuito identifica, através de sensores infravermelhos dispostos nas extremidades das esteiras, se um objeto foi inserido ou retirado da esteira, informando ao computador através da porta paralela. O sistema identifica quando há objetos sobre a esteira de tal forma que o circuito mantenha o motor ligado.

Desta forma, é possível a automação e controle do conjunto de esteiras. Na tela do computador é exibido o total de objetos que já passaram sobre as esteiras e

a quantidade de objetos que ainda estão sobre elas. Na figura 1 é mostrado o fluxograma para controle das esteiras.

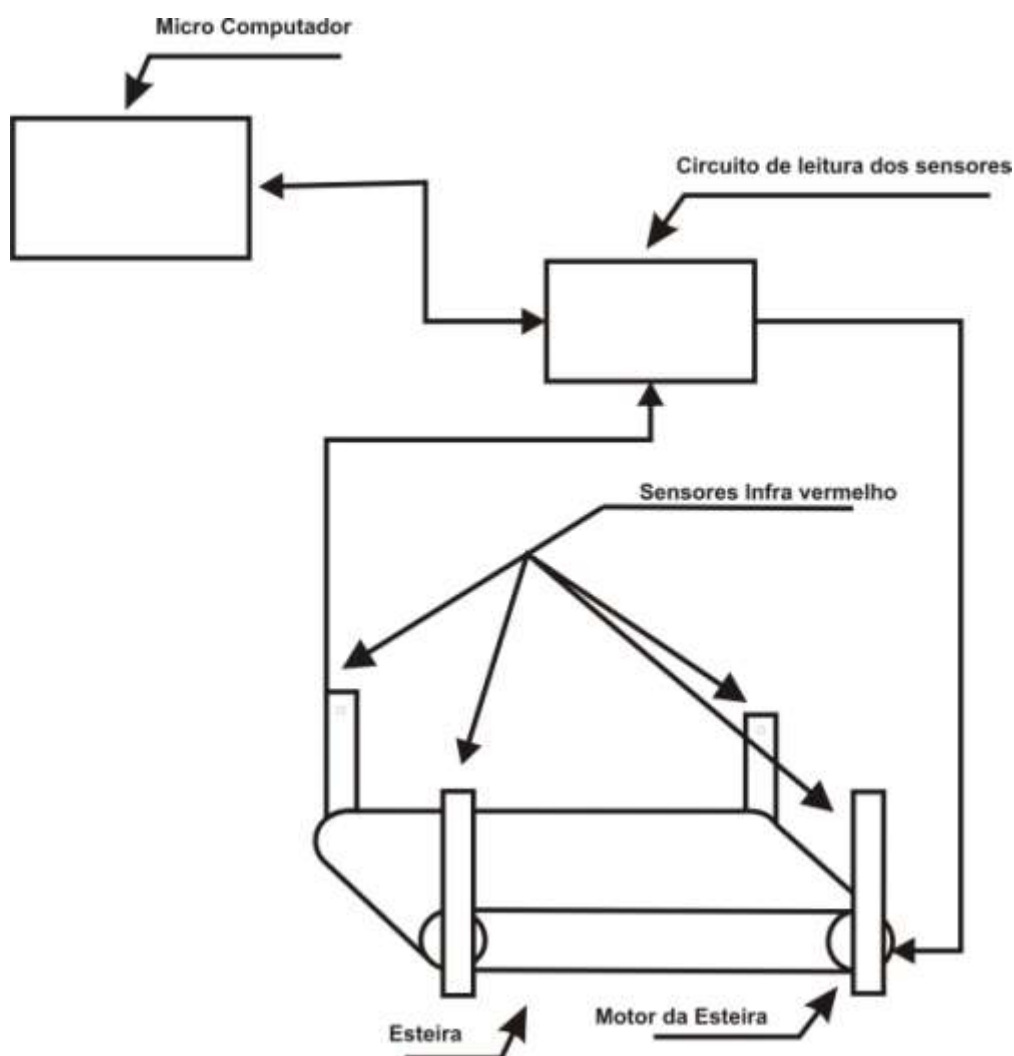


Figura 2 Fluxograma de controle das esteiras.

1.1 Motivação

A escolha foi baseada em observações e no estudo dos projetos dos colegas de semestres anteriores. Após a pesquisa no projeto da Juliana Ferreira Carrijo [2], foi dada continuidade ao seu tema, porém utilizando outros recursos para implementar uma proposta de solução para economia e conservação de esteiras automatizadas.

As principais diferenças do projeto da Juliana é que a proposta dela trabalha com microcontrolador e controla apenas uma esteira, neste trabalho proponho utilizar o computador com comunicação pela porta paralela e controlar mais de uma esteira.

Os mecanismos utilizados nas esteiras elétricas estão presentes desde os setores primários da economia como em mineradoras e agricultura, até setores fins como no comércio que utilizam esteiras nos caixas de supermercados e escadas rolantes.

O resultado desejado com este projeto é apresentar uma proposta de automação para esteiras rolantes, com a finalidade de economia de energia e de diminuir o desgaste dos componentes das esteiras, uma vez que as mesmas deixarão de funcionar sem a existência de objetos sobre as mesmas gerando economia de energia.

1.2 *Objetivo geral.*

O principal objetivo deste projeto é desenvolver e automatizar um conjunto de esteiras rolantes acionadas apenas quando for detectada a necessidade. Para isso, será utilizado um circuito interligado ao computador, um motor de corrente contínua e dois sensores infravermelho. O programa utilizado para efetuar o controle de todos os dispositivos mencionados será desenvolvido utilizando a linguagem de programação *Java*.

1.3 Objetivos específicos.

São objetivos específicos do projeto:

Montar um circuito que controle sete esteiras que utilizam motor de corrente contínua, sensor infravermelho, outros dispositivos necessários e comunique-se com o computador através da porta paralela.

Desenvolver o programa de controle que conte quantos objetos tem sobre as esteiras e o total que passou pela mesma utilizando a linguagem *Java*.

Montar uma maquete de uma esteira que implemente a proposta e demonstre o funcionamento para uma esteira do projeto.

Demonstrar através de simulação com LED's e chaves o funcionamento das 6 esteiras não demonstradas com a maquete.

O número de sete esteiras foi escolhido, mesmo verificando que o modelo escolhido para esta solução comportaria oito esteiras tranquilamente, pelo motivo de ter um bit sobrando na saída da porta paralela do computador para utilizar como retorno de informação, caso seja necessário.

1.4 Organização da monografia.

Capítulo 1: Apresenta a idéia geral do projeto, os objetivos a serem alcançados, uma visão geral do que será desenvolvido na monografia com relação a parte física e também do programa de controle.

Capítulo 2: Descreve as fases de construção da maquete e do programa de controle desenvolvido.

Capítulo 3: São explanadas as considerações finais a respeito do desenvolvimento do projeto como um todo, detalhando as dificuldades encontradas,

as conclusões e sugestões para trabalhos futuros. Neste capítulo também é apresentado um item com os resultados obtidos através de testes efetuados após a construção da maquete e a finalização do programa desenvolvido.

Capítulo 2. Desenvolvimento do protótipo

Em linhas gerais, o funcionamento deste protótipo consiste em construir um circuito que transforme os sinais enviados por cada um dos quatorze sensores IR Infravermelho, dois para cada esteira, em uma seqüência de cinco bits. Envia-se esse sinal para o computador pela porta LPT1. No computador um programa escrito em *Java* identifica se o sinal enviado é de um sensor de entrada ou de saída e de qual esteira ele faz parte, se esta esteira tem ou não objetos sobre ela, com essas informações o sistema decide se o motor da esteira deve estar ligado ou desligado, escrevendo esta resposta na porta LPT1, o circuito então liga ou desliga o motor da respectiva esteira.

O diagrama de blocos abaixo demonstra a idéia do projeto.

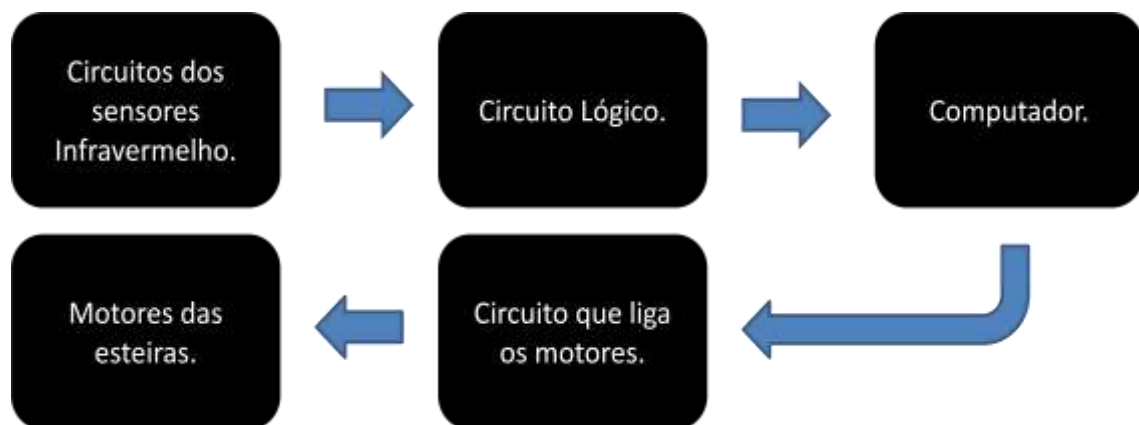


Figura 3 Diagrama de bloco do projeto.

Para demonstrar o projeto para a banca foi construído o circuito que recebe o sinal dos quatorze sensores IR, uma esteira e seus dois sensores IR e o sistema em *Java* para controle de sete esteiras. As seis esteiras restante serão demonstradas através de simulação utilizando uma maleta laboratório modelo “Minipa 500 em um”

de eletrônica, com LED's simulando os motores e chaves simulando os sensores IR destas esteiras.



Figura 4 Maleta laboratório modelo “Minipa 500 em um”

A seguir são descritas todas as conexões que compõem esta estrutura e a construção do protótipo.

2.1 Tabela Verdade criada para o projeto.

Para desenvolver a tabela verdade do projeto, primeiro foi determinado o total de possibilidades de entradas permitidas para o sistema. Representados em cada linha da tabela, estas entradas são na pratica a saída de cada circuito que interpreta

cada um dos quatorze sensores IR, representados na tabela como SX onde $1 \leq X \leq 14$ sendo que para cada esteira é utilizado dois sensores um para entrada outro para a saída. Os valores de X impares representam os sensores de entrada de objetos nas esteiras, e os valores X pares os de saída, a tabela 1 ilustra esta explicação.

Esteira	Sensor de entrada	Sensor de saída
Esteira 1	S1	S2
Esteira 2	S3	S4
Esteira 3	S5	S6
Esteira 4	S7	S8
Esteira 5	S9	S10
Esteira 6	S11	S12
Esteira 7	S13	S14

Tabela 1 Distribuição IR x Esteira.

A saída da tabela foi padronizado como BX onde $1 \leq X \leq 5$, que na pratica representam os 5 bits de entrada na porta LPT1. Estes bits foram separados da seguinte forma dois bits para informar se o sinal é de saída ou entrada na esteira e três bits para endereçar as sete esteiras, o bit B1 ativo informa a entrada das esteiras o bit B2 informa a saída das esteiras e os bits B3, B4 e B5 endereçam as sete esteiras com seqüências pré definidas. Desta forma é possível informar ao programa *Java* em qual esteira um objeto entrou ou saiu.

A tabela abaixo representa a tabela verdade utilizada no projeto com todas as suas entradas e saídas.

Entradas														Saídas				
S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	B1	B2	B3	B4	B5
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1
0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0

Tabela 2 Tabela Verdade do projeto

Para esta proposta foi padronizado que, é permitido apenas uma entrada de dados por vez, por este motivo cada linha da tabela só recebe um valor 1.

2.2 Álgebra de Boole utilizada no projeto.

Para elaborar a equação booleana das esteiras, a tabela verdade foi separada em cinco equações, cada equação tem como resultado um dos 5 bits de saída do circuito lógico do projeto.

As equações booleanas do circuito lógico da esteira são equações do tipo ou – exclusivo. A porta lógica para estas equações é a XOR, esta porta tem a função de subtração lógica. A função básica do circuito lógico é subtrair as entradas que estão inativas desta forma é possível com uma entrada ativar a sequência correta de bits.

O bit B1 tem sinal lógico um, quando qualquer entrada impar for acionada, informando a entrada de um objeto em uma das esteiras.

O bit B2 tem sinal lógico um, quando qualquer entrada par for acionada, informando a saída de um objeto em uma das esteiras.

O bit B3 tem sinal lógico um, quando apenas uma das entradas S9, S10, S11, S12, S13 ou S14 é acionada.

O bit B4 tem sinal lógico um, quando apenas uma das entradas S5, S6, S7, S8, S13 ou S14 é acionada.

O bit B5 tem sinal lógico um, quando apenas uma das entradas S3, S4, S7, S8, S11 ou S12 é acionada.

A tabela abaixo com a seqüência binária das saídas B3, B4, e B5, exibe melhor a seqüência binária que identifica cada esteira.

B3	B4	B5	Esteira
0	0	0	Esteira 1
0	0	1	Esteira 2
0	1	0	Esteira 3
0	1	1	Esteira 4
1	0	0	Esteira 5
1	0	1	Esteira 6
1	1	0	Esteira 7

Tabela 3 Endereçamento das Esteiras

Após a separação das saídas obtêm-se as cinco equações booleanas abaixo.

$$(S1 \oplus S3) \oplus (S5 \oplus S7) \oplus (S9 \oplus S11) \oplus S13 = B1 \quad (1)$$

$$(S2 \oplus S4) \oplus (S6 \oplus S8) \oplus (S10 \oplus S12) \oplus S14 = B2 \quad (2)$$

$$(S9 \oplus S10) \oplus (S11 \oplus S12) \oplus (S13 \oplus S14) = B3 \quad (3)$$

$$(S5 \oplus S6) \oplus (S7 \oplus S8) \oplus (S13 \oplus S14) = B4 \quad (4)$$

$$(S3 \oplus S4) \oplus (S7 \oplus S8) \oplus (S11 \oplus S12) = B5 \quad (5)$$

2.3 Circuito Elétrico utilizada no projeto.

O circuito elétrico da esteira é dividido em três blocos básicos. O primeiro bloco é composto por quatorze circuitos, dois para cada esteira, que geram o sinal do sensor IR e informam ao circuito lógico. O segundo bloco é o circuito lógico que recebe o sinal da primeira parte, converte na sequência de cinco bits e escreve na porta LPT1 do computador. O terceiro bloco é o circuito que liga os motores, este recebe do programa em *Java* a resposta de qual motor deve ser ligado ou desligado e efetua esta tarefa.

A figura 5 mostra o circuito que foi montado para o projeto com os 3 blocos marcados.

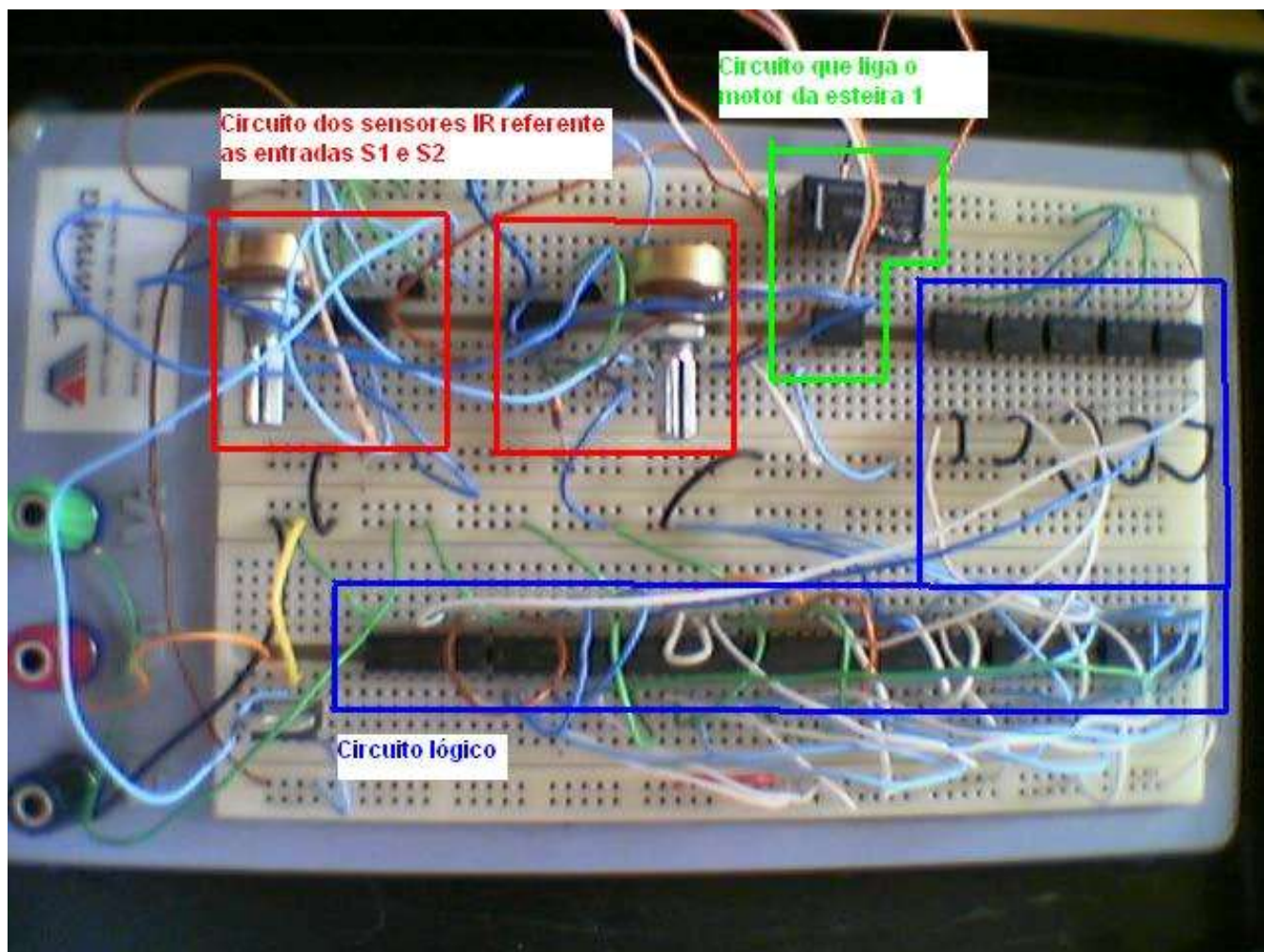


Figura 5 Circuito montado para o projeto

2.3.1 Circuito elétrico dos sensores IR.

Como mencionado no item 2.3, a primeira parte do circuito elétrico do projeto são os quatorze circuitos que geram o sinal do sensor IR. Este circuito é formado basicamente por um LED IR, um fotodiodo, um amplificador utilizando um transistor NPN e um amplificador operacional 741, que é montado como um comparador de tensão, que é uma das formas de montar esse componente. Esse tem duas entradas. Uma entrada "+" para porta não inversora e uma "-" para a inversora. Quando a entrada "+" possui uma diferença de mV maior que a entrada "-", a saída se torna alta, caso contrário a saída é baixa. Os demais componentes como

resistores e regulador de tensão são para completar a estrutura de montagem do circuito.

O desenho exibido na Figura 6 é o esquema elétrico para um dos 14 circuitos que geram o sinal dos sensores IR, este desenho foi desenvolvido no software Multisim Power Pro Edition, versão 10.0.144, da National Instruments.

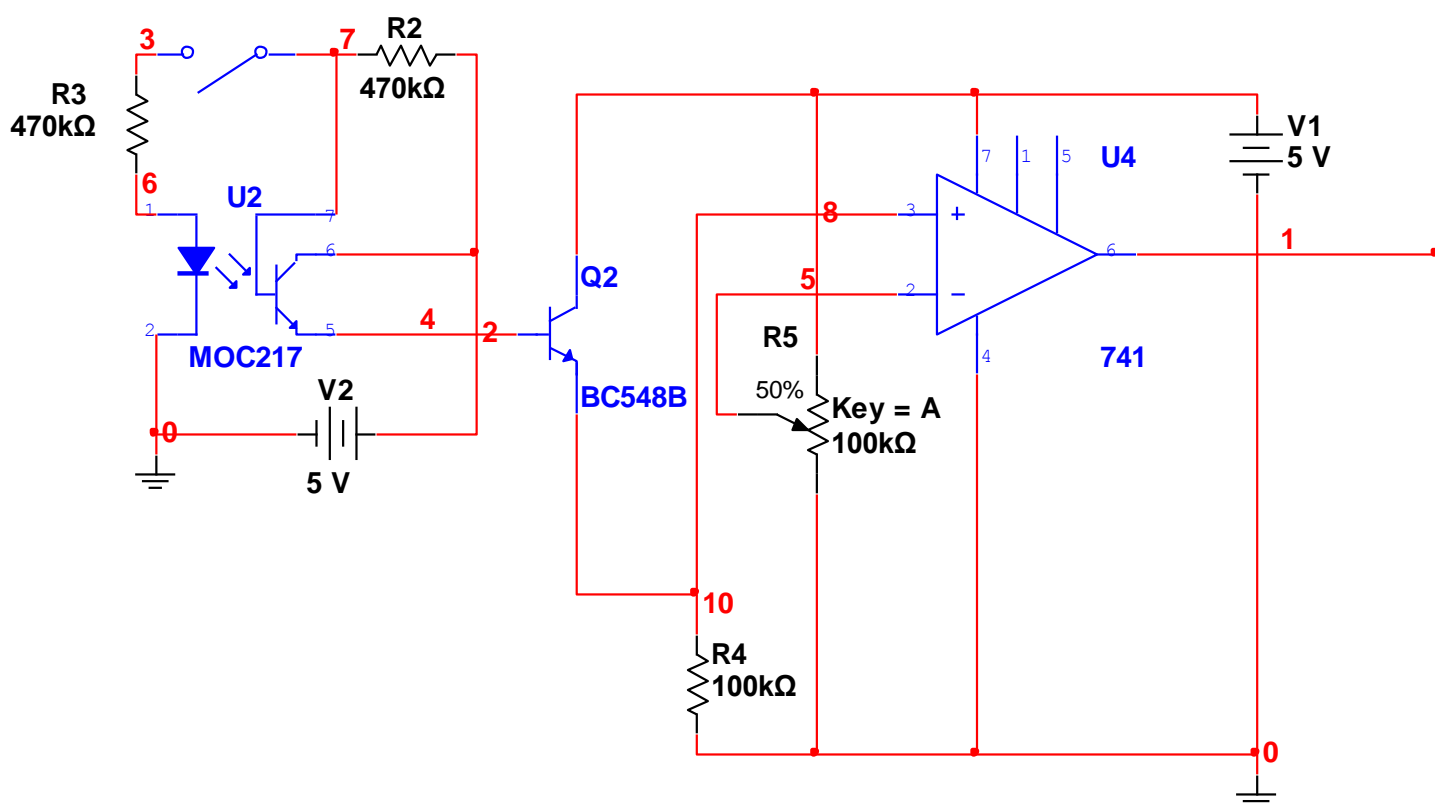


Figura 6 esquema elétrico do circuito que gera o sinal dos 14 sensores IR

A tensão que o fotodiodo envia é amplificada pelo transistor NPN e enviado ao amplificador operacional 741. O 741 compara a diferença de tensão entre o sinal do amplificador e um sinal que vem de um regulador de tensão, aplicando em sua saída uma tensão de 4.5V ou 1V.

Enquanto o sinal do LED IR não é cortado, o sinal enviado ao circuito lógico pelo amplificador operacional 741 é de 4,5 V interpretado com um sinal lógico 1. Quando algum objeto passa entre o fotodiodo e o LED IR ele corta o sinal e o 741 envia um sinal de menos de 1V que é interpretado como sinal lógico 0 pelo circuito lógico.

Para a elaboração da Figura 6 no simulador, foi substituído o fotodiodo e o LED IR pelo acoplador ótico MOC217, e o circuito necessário para ligá-lo, desta forma fica mais pratico a elaboração no software de simulação e não influência em nada o funcionamento do circuito.

2.3.2 Projeto das portas lógicas utilizado no projeto.

A segunda parte do circuito utilizado no projeto é o circuito das portas lógicas. Para desenhar este circuito foram aplicadas as equações booleanas descritas no item 2.2, de forma que a seqüência de cinco bits seja obtida como resultado da execução física das mesmas.

O componente utilizado que implementa a porta lógica XOR é o CI (Circuito Integrado) DM74LS86, que possui internamente quatro portas lógicas. [5]

Para implementar a parte lógica do circuito do projeto foram utilizados vinte e cinco portas lógicas, em sete CI's DM74LS86. Como cada CI possui quatro portas, ficam sobrando três portas lógicas em um deles.

O desenho exibido na Figura 7 é o esquema lógico do circuito de controle das sete esteiras este desenho foi desenvolvido no software Multisim Power Pro Edition, versão 10.0.144, da National Instruments.

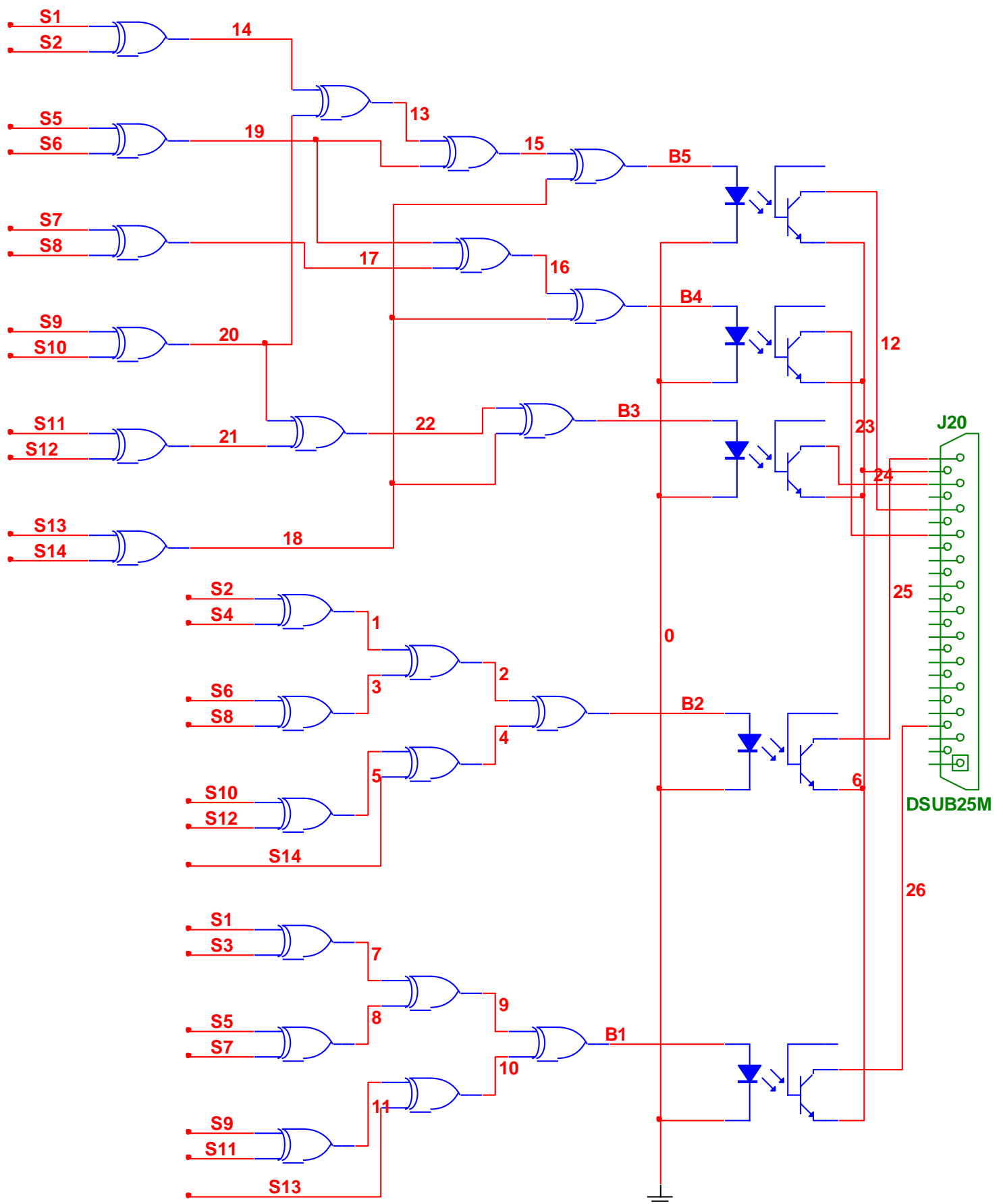


Figura 7 Circuito lógico do projeto

O circuito lógico do projeto tem 14 entradas que são as SX onde $1 \leq X \leq 14$ e são as mesmas entradas representadas na tabela 2 da mesma forma tem 5 saídas as BX onde $1 \leq X \leq 5$.

Cada um dos 14 circuitos descritos no item 2.3.1 tem a sua saída ligada a uma das entradas SX. A seqüência de portas XOR executa uma das equações descritas no item 2.2 conforma a entrada SX acionada, obtendo um resultado de 5 bits, que é a saída do circuito lógico.

Os resultados obtidos com as operações lógicas são os valores cadastrados no computador e obedece a seqüência da tabela abaixo.

Esteira	Valor de entrada de objetos em binário	SX impar acionado na entrada.	Valor de saída de objetos em binário	SX par acionado na entrada.
Esteira 1	10000	S1	01000	S2
Esteira 2	10001	S3	01001	S4
Esteira 3	10010	S5	01010	S6
Esteira 4	10011	S7	01011	S8
Esteira 5	10100	S9	01100	S10
Esteira 6	10101	S11	01101	S12
Esteira 7	10110	S13	01110	S14

Tabela 4 Tabela explicativa do circuito lógico.

Cada saída do circuito lógico está ligada a porta LPT1 do computador através de um acoplador ótico, esta ligação é descrita em detalhes no item 2.5.

2.3.3 Circuito elétrico que liga os motores das esteiras.

O circuito elétrico que liga os motores das esteiras é a terceira parte do circuito principal, está ligado à porta LPT1 do computador (pinos de D0 a D6), esta ligação está descrita com maiores detalhes no item 2.5. Cada saída da porta LPT1 refere-se ao motor de uma esteira.

Quando uma das saídas da LPT1 está com sinal lógico 1, o circuito aciona um relê que fecha um circuito de 12V ligando o motor da esteira correspondente a saída que foi acionada.

O desenho exibido na Figura 8 é o esquema elétrico do circuito que liga os motores das esteiras, este desenho foi desenvolvido no software Multisim Power Pro Edition, versão 10.0.144, da National Instruments.

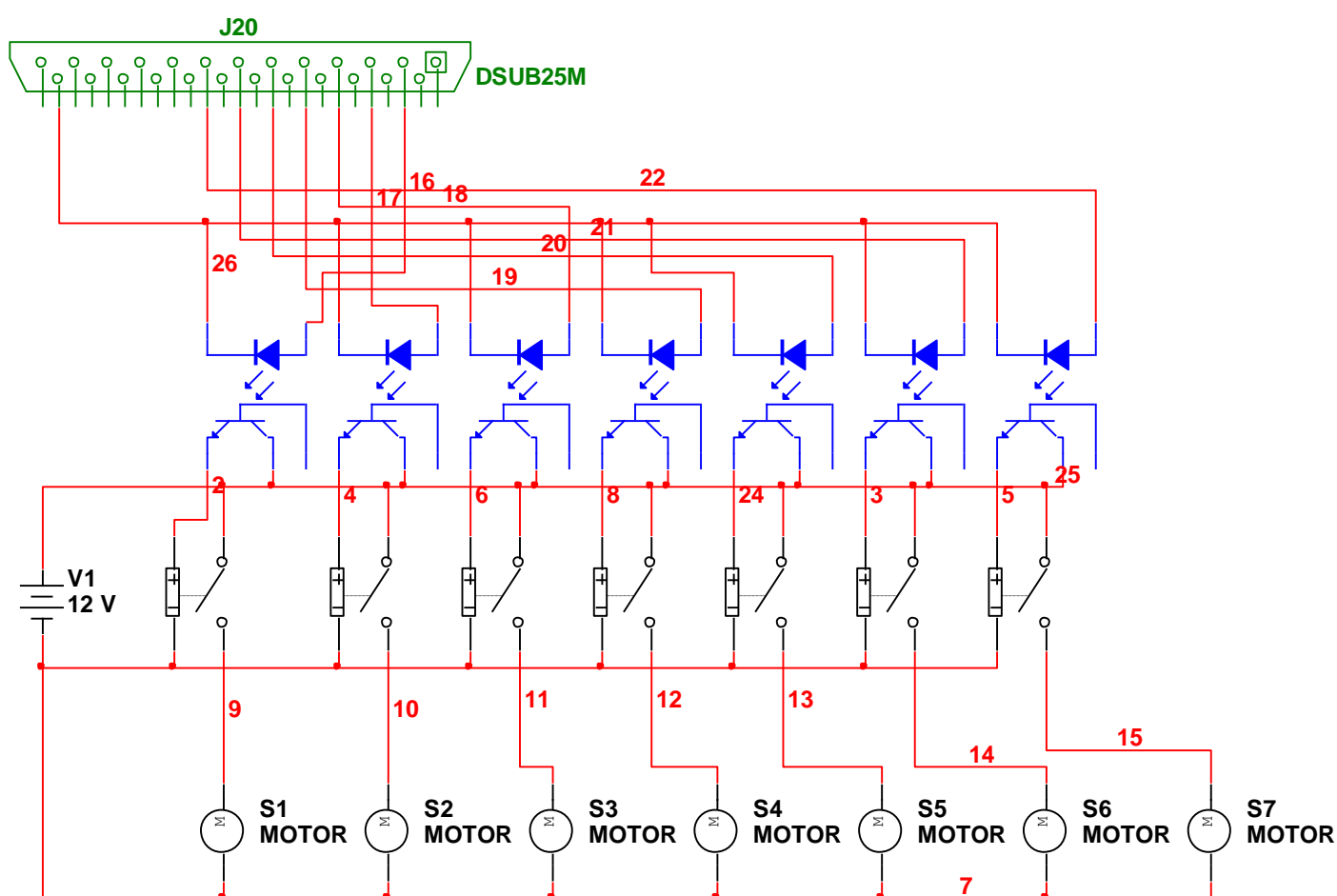


Figura 8 Circuito que liga os motores das esteiras

2.4 *Circuito completo utilizado no projeto.*

O circuito completo para as sete esteiras possui quatorze circuitos, como descrito no item 2.3.1. Cada um destes tem sua saída à entrada correspondente no circuito lógico descrito no item 2.3.2, o circuito lógico converte estas entradas em cinco bits executando as operações booleanas descritas no item 2.2. Estes bits são repassados para a porta LPT1 do computador para que o programa em *Java* possa ler e interpretar.

O esquema elétrico completo do circuito para as sete esteiras está exibido no Apêndice A.

2.5 *Ligação do circuito utilizado no projeto com a porta LPT1 do computador.*

A porta do computador LPT1 possui o mapeamento exibido na figura 9, considerando o conhecimento prévio da sequência dos pinos, seus respectivos nomes e o funcionamento da porta paralela do computador, neste item é explicada a ligação do circuito lógico com a porta LPT1.

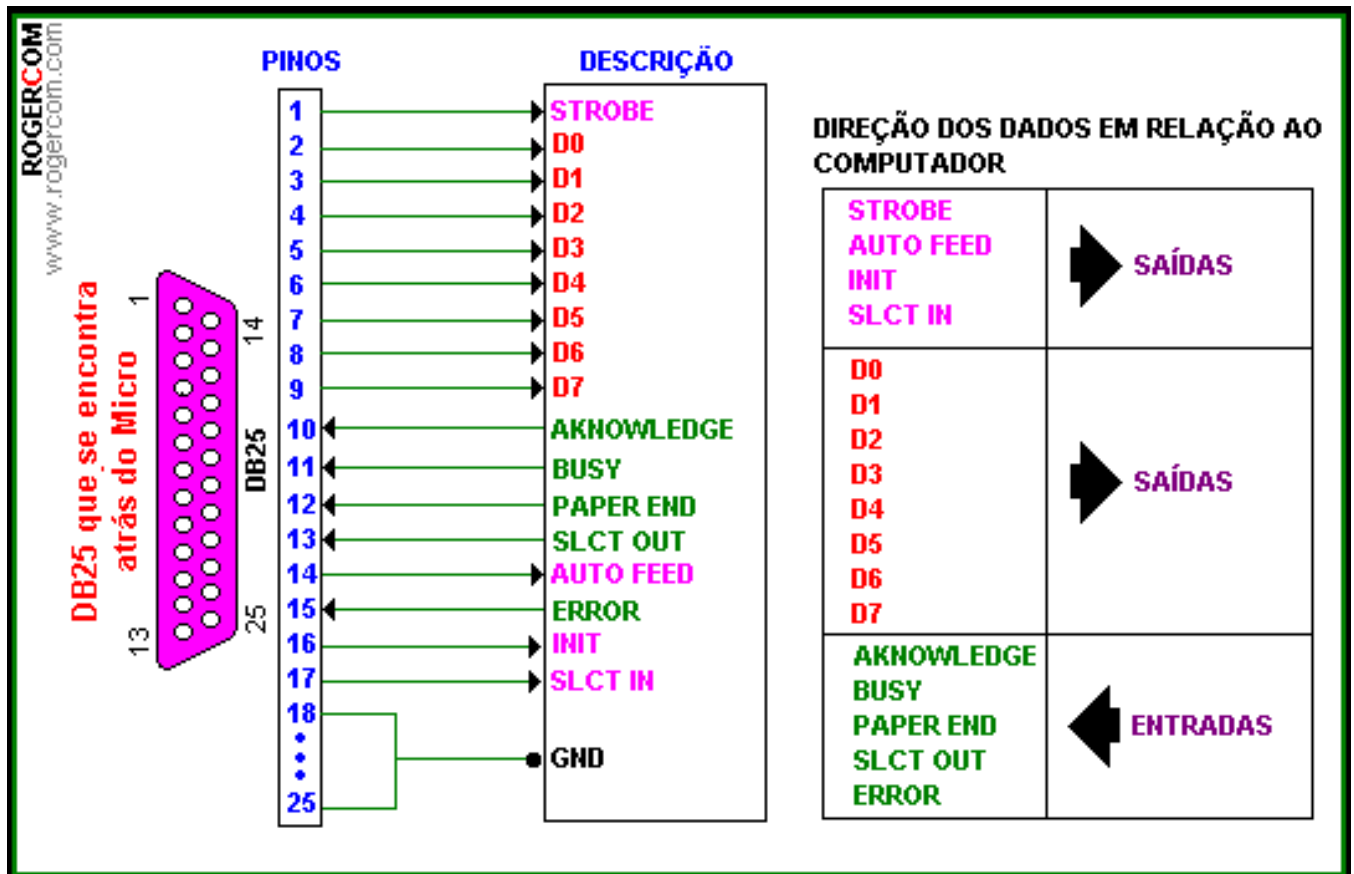


Figura 9 Esquema de funcionamento da entrada paralela do computador com o conector DB25 no modo SPP. [19]

A porta LPT1 possui cinco pinos de entrada conforme exibido na Figura 9. Estes pinos estão carregados por padrão com 4,5V que o computador interpreta como valor lógico 0, observando que o pino 11 (BUSY) tem o sinal interpretado de forma invertida, o valor 4.5V refere-se ao valor lógico 1 para este pino. Para escrever o valor lógico 1 no pino é necessário fechar o circuito entre o respectivo pino e um dos pinos de 18 a 25 que são os GND (Terra do circuito) da porta. Para fazer esta função foi colocado o acoplador ótico 4N25, de forma a proteger a porta LPT1 do computador de possíveis descargas de maior tensão.

O acoplador ótico 4N25 internamente é feito basicamente de um LED e um transistor ótico, quando o LED interno é aceso, o transistor ótico fecha o circuito

permitindo a passagem de carga entre os pinos 4 e 5 do componente. Desta forma, o circuito que acende o LED fica totalmente protegido do circuito que está ligado ao transistor ótico.

Para a ligação entre o circuito lógico do projeto e a porta LPT1 do computador, cada pino da porta LPT1 ficou ligada a um acoplador ótico 4N25 em seus transistores óticos e o circuito lógico do projeto acendia o LED interno do 4N25.

No projeto a saída do circuito lógico é ligada aos pinos da porta LPT1 conforme Tabela 5 abaixo.

Saída do Circuito	B1	B2	B3	B4	B5
LPT1	15	13	12	10	11

Tabela 5 Mapeamento Circuito Logico X Porta LPT1

A porta LPT1 também está ligada aos circuitos que ligam o motor de cada esteira conforme descrito no item 2.3.3. Da mesma forma como foi utilizado o 4N25 na entrada de dados, ele foi utilizado na saída de dados da porta LPT1, porém desta vez de forma inversa a porta LPT1 liga os LED`s internos, e o transistor ótico aciona o relé que liga o motor da respectiva esteira. O mapeamento das esteiras e dos pinos da porta LPT1 é exibido na Tabela 6 abaixo.

Pino LPT1	2	3	4	5	6	7	8
Esteira	1	2	3	4	5	6	7

Tabela 6 Mapeamento porta LPT1 X Esteiras

Conforme exibido na Tabela 6 cada esteira está ligada a um pino de saída da porta LPT1. O programa em *Java* ativa ou desativa o pino e por ligação vai controlar a esteira correspondente.

Para o programa *Java* ligar ou desligar uma esteira ele escreve na porta LPT1 os valores conforme a Tabela 7 exibida abaixo.

Valor de saída do Java	Esteira	Valor Binário escrito na LPT1
1	Esteira 1	0000001
2	Esteira 2	0000010
4	Esteira 3	0000100
8	Esteira 4	0001000
16	Esteira 5	0010000
32	Esteira 6	0100000
64	Esteira 7	1000000

Tabela 7 Saída do programa Java

2.6 Programa Java que controla as esteiras.

O programa em Java controla a quantidade de objetos que está encima das esteiras e liga ou desliga a mesma, para isso ele segue a lógica do diagrama de bloco exibido na Figura 10.

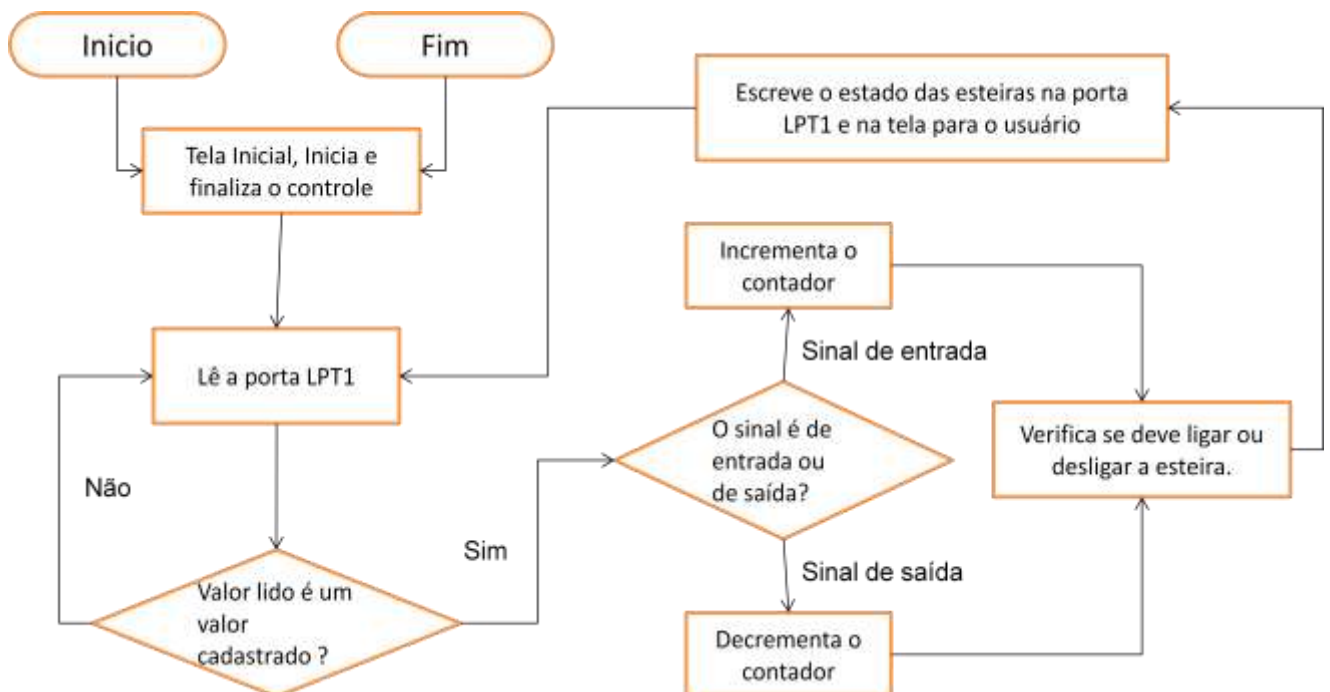


Figura 10 Diagrama de blocos do programa Java.

O programa tem uma tela inicial exibido na Figura 11

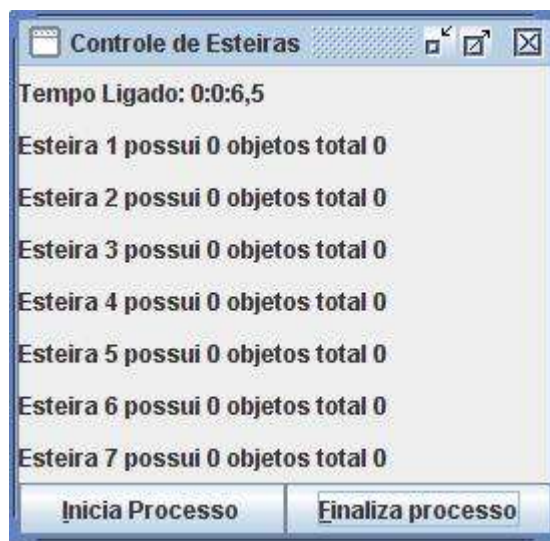


Figura 11 Tela do programa *Java*

Nesta tela o usuário liga ou desliga o processo de controle e nela também é exibido a quantidade de objetos que está sobre cada esteira e o total de objetos que passou sobre a mesma.

Após o início do processo o programa inicia um loop que fica lendo a porta LPT1, todo o valor que é lido na porta é verificado, caso este valor não esteja cadastrado ele é ignorado, quando está cadastrado, é verificado se é de um sensor de entrada ou de saída, caso seja um sinal de entrada é incrementado um contador caso um sinal de saída este contador é decrementado, o próximo passo é verificar se o contador está zerado, pois isso indica que não há objetos encima da respectiva esteira, para o contador > 0 e o estado da esteira desligada o programa escreve na porta LPT1 para ligar o motor, caso contrario, o contador ≤ 0 e o estado ligado o programa desliga o motor. A tabela 8 descreve a resposta do programa para as diversas possibilidades que o projeto prevê.

Contador	Estado da esteira	Resposta do Programa
≤ 0	Ligada	Desliga
≤ 0	Desligada	Mantêm
> 0	Ligada	Mantêm
> 0	Desligada	Liga

Tabela 8 Contador X Estado da esteira X Resposta do Programa

Após esta verificação o programa escreve na porta LPT1 e na tela do usuário o resultado e reinicia o processo da leitura da porta LPT1.

O programa *Java* que controla as esteiras é constituído basicamente de três classes:

ConfigSistema.java. Esta classe contém as variáveis fixas do programa. É nesta classe também que são cadastrados os valores que serão lidos na porta LPT1, que representa cada um dos 14 sensores das esteiras.

O código fonte da classe *ConfigSistema.java* está exibido no Apêndice B.

Tela.java. Esta classe desenha a tela e disponibiliza os botões para ligar e desligar o controle das esteiras.

O código fonte da classe *Tela.java* está exibido no Apêndice C.

ThreadLoop.java. Esta classe é que faz praticamente todas as funções do programa. Ela fica lendo a porta LPT1 e comparando com os valores pré estabelecidos da classe *ConfigSistema.java* para cada sensor. Caso o valor lido na porta seja um dos valores estabelecidos é incrementado ou decrementado a conta da esteira correspondente, conforme descrito acima.

O código fonte da classe *ThreadLoop.java* está exibido no Apêndice D

.

Capítulo 3. Considerações finais.

Neste capítulo são apresentadas as principais dificuldades encontradas no decorrer da execução do projeto, as conclusões obtidas com o seu desenvolvimento e sugestões para trabalhos futuros.

3.1 Resultados obtidos.

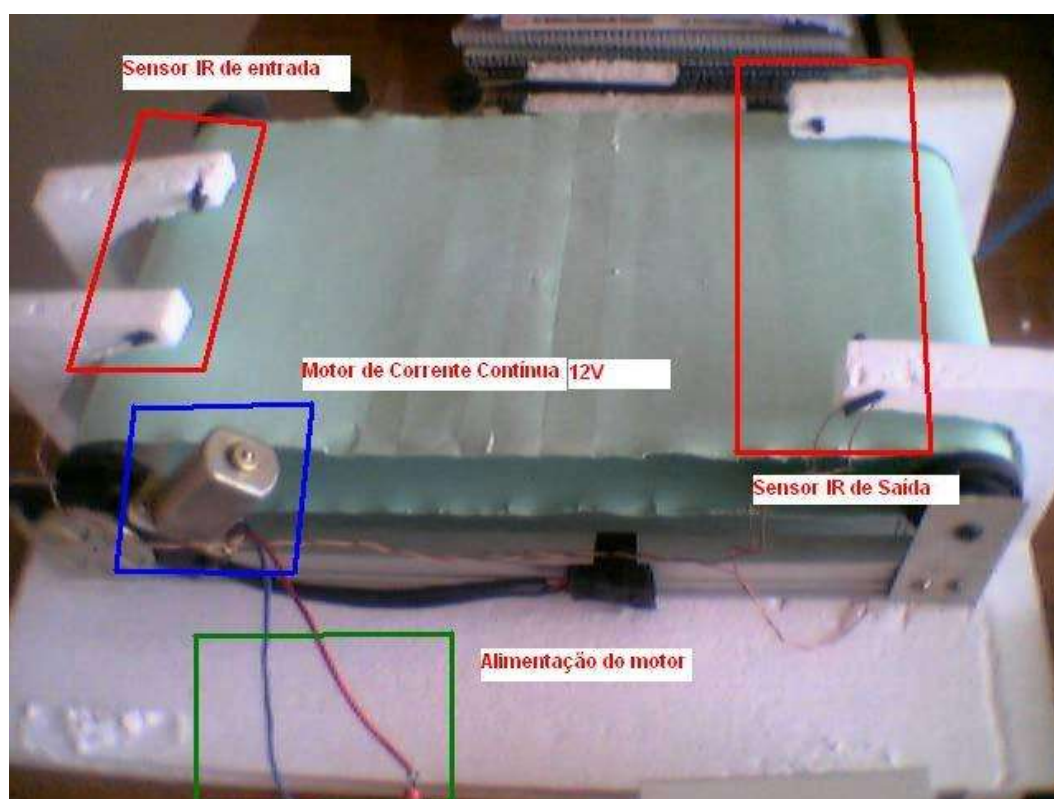


Figura 12 Protótipo da esteira



Figura 13 Circuito de controle e maleta Minipa 500 em um

Após a construção da estrutura da esteira, do circuito elétrico e lógico e desenvolvimento do programa, obteve-se o funcionamento da esteira e a simulação de controle de seis esteiras não montadas, porém foram verificados alguns problemas. O primeiro foi devido à falta de gabarito na construção da esteira o fotodiodo e o LED IR tiveram certa dificuldade para ficar alinhados. Para solucionar esse problema o sensor IR foi montado em uma estrutura separada da esteira. Também devido à falta de gabarito na construção da esteira os tubos por onde passa o papel da esteira não ficaram alinhados. Por esse motivo o papel após alguns instantes se desloca para uma das extremidades e sai da esteira. Devido à abrangência do projeto acadêmico, esses não são problemas graves.

Outro problema é que quando dois ou mais sensores são acionados ao mesmo tempo, o sinal enviado ao computador não é reconhecido como um sinal válido e

não executa tarefa alguma, ou seja, não incrementa e nem decrementa o contador das esteiras. Para solucionar esse problema, uma proposta seria utilizar uma memória na saída de cada circuito gerador do sinal IR que armazenaria o último sinal enviado, e um ciclo para a leitura destes circuitos, o circuito lógico iria ler esta memória em uma seqüência ao invés de receber o sinal do circuito do IR, impedindo assim que mais de um circuito envie sinal ao mesmo tempo.

3.2 *Dificuldades encontradas.*

Neste projeto a principal dificuldade encontrada foi a falta de ferramentas adequadas para a construção da maquete, pois como esta foi construída pelo autor sem muitos gastos, com partes de diversos objetos, entre eles um patins e uma porta de alumínio, a construção ficou sem muito gabarito, gerando alguns problemas no seu funcionamento.

Como o autor tem experiência em construção de hardwares e softwares a implementação do circuito e do programa em *Java* não teve grandes dificuldades.

Todos os componentes utilizados podem ser encontrados no mercado de Brasília sem muitas dificuldades.

3.3 *Conclusões.*

Este projeto teve como objetivo a automação de sete esteiras rolantes através do computador. O objetivo era aplicar conhecimentos por meio do desenvolvimento de todo o projeto. Tomando como base os objetivos específicos listados no item 1.3 desta monografia, chega-se a conclusão que o objetivo foi alcançado.

Quanto ao objetivo de montar o circuito que interprete o sensor infravermelho e repasse essa informação para a porta LPT1 do computador, esta tarefa foi executada com uma pequena ressalva, no caso de mais de um sensor ser acionado ao mesmo tempo. Este problema pode ser solucionado com a proposta sugerida no item 3.1, porém foi demonstrando que receber uma informação de um circuito e repassar esta informação para o computador é uma tarefa relativamente simples e prática.

Com relação ao objetivo específico de desenvolver o programa de controle na linguagem *Java*, este objetivo foi completamente alcançado.

O objetivo específico de controlar a quantidade o objetos que passaram e estão sobre a esteira também foi completamente alcançado.

3.4 Sugestões de trabalhos futuros.

A partir da monografia e do projeto desenvolvido é possível sugerir algumas evoluções nos estudos efetuados, que são descritas abaixo para implementações em projetos futuros:

Implementar uma memória para o circuito de leitura do IR e ciclo de leitura destas memórias, para solucionar o problema de ter mais de uma esteira enviando sinais simultaneamente no sistema;

Converter a saída do circuito lógico para porta serial ao invés de paralela, pois esta porta esta ficando obsoleta e dessa forma pode-se controlar uma quantidade maior de esteiras;

Converter a saída do circuito lógico para rede sem fio, podendo assim controlar uma quantidade maior de esteiras e a uma distância maior também.

Implementar a variação de velocidade no motor de corrente contínua, como forma de poder controlar esta velocidade quando houver uma demanda maior da esteira, implementando rotinas pré-definidas para a esteira.

Implementar o projeto utilizando um microcontrolador.

Realizar estudos e pesquisas para mensurar a economia de energia e desgastes das peças do equipamento alcançado por meio do uso deste projeto.

3.5 Custo estimado do projeto.

O custo real do projeto não pode ser estimado com precisão, pois diversos componentes não foram comprados, muitas ferramentas o autor do projeto já tinha, abaixo segue a lista dos componentes e materiais comprados.

Nome/Descrição do Dispositivo Adquirido	Custo do Dispositivo
LED 5MM alto brilho (2 unidades)	R\$ 2,60
Conectores DB25 (2 tipo Fêmea e 1 tipo Macho)	R\$ 4,65
Conector fêmea RJ 45	R\$ 1,80
Conector fêmea RJ 11	R\$ 0,60
Fonte estabilizada 12V 3.5A	R\$ 71,64
LED fotodiodo (2 unidades)	R\$ 3,00
LED emissor Infravermelho (2 unidades)	R\$ 3,00
Rele 12V 10A	R\$ 2,50
Caixa plástica PB 900	R\$ 26,50
Cabo Flet 16 vias (1M)	R\$ 4,50
Custo Total Estimado	R\$ 120,79

Tabela 9 Custo estimado do projeto

Referências bibliográficas.

- [1] - BRAGA, Newton C., Saber Eletrônica Especial Nº 8 – 2002, Curso de Eletrônica Digital.
- [2] - CARRIJO, Juliana Ferreira, Esteira Rolante Microcontrolada. Brasília, julho 2008. 99 p. Monografia (Graduação em Engenharia da Computação). Centro Universitário de Brasília, Faculdade de Tecnologia e Ciências Sociais Aplicadas.
- [3] - CLOSE, Charles M., Circuitos Lineares, 2ª Edição: Rio de Janeiro: LTC, 1975.
- [4] - DAGHLIAN, Jacob, Lógica e Álgebra de Boole, 4ª Edição: São Paulo, Atlas, 1995
- [5] - DM74LS86 Quad 2-Input Exclusive-OR Gate
- [6] - FERREIRA, Aurélio Buarque de Holanda. Minidicionário da língua portuguesa. 3ª ed. Rio de Janeiro: Nova Fronteira, 1993.
- [7] - FITZGERALD, A. E.; Charles Kingsley Jr.; Stephen D. Umans. Máquinas Elétricas. 6ª ed. Porto Alegre: Bookman, 2006.
- [8] - FITZGERALD, A. E.; Charles Kingsley Jr; Alexandre Kusko. Máquinas Elétricas. Ed. McGraw-Hill, 1975
- [9] - GILIO, Aluisio Simone. Máquinas de Corrente Contínua. 1ª Ed. São Paulo: Érica, 1998.
- [10] - <http://br.geocities.com/saladefisica7/funciona/motoreletrico.htm>
Acesso: 23 de novembro de 2008
- [11] - http://equiplast-sc.com.br/site/index.php?option=com_rsgallery2&Itemid=27&catid=1
Acesso: 08 de novembro de 2008
- [12] - <http://informatica.hsw.uol.com.br/portas-paralelas2.htm>
Acesso: 05 de outubro de 2008
- [13] - <http://java.sun.com/>
Acesso: 23 de setembro de 2008
- [14] - <http://java.sun.com/javase/6/docs/technotes/guides/jni/>
Acesso: 23 de setembro de 2008
- [15] - <http://www.clubedohardware.com.br/duvidas/64>
Acesso: 23 de setembro de 2008

[16] - <http://www.dicweb.com/bb.htm>

Acesso: 10 de outubro de 2008

[17] - <http://www.geocities.com/Juanga69/parport/>

Acesso: 08 de outubro de 2008

[18] - <http://www.jobtecltda.com.br/paralela/paralela.htm>

Acesso: 05 de outubro de 2008

[19] - <http://www.rogercom.com/pparalela/introducao.htm>

Acesso: 23 de setembro de 2008

[20] – IRWIN, J. David, Análise de Circuitos em Engenharia, 4ª Edição: São Paulo, Makron Books, 2005

[21] - MARKUS, Otávio. Circuitos Elétricos: Corrente Contínua e Corrente Alternada. São Paulo: Érica, 2001.

[22] - THOMAZINI, Daniel; Pedro Urbano B. de A.. Sensores Industriais. 1ª ed. São Paulo: Érica, 2005.

[23] - TORO, Vincent Del. Fundamentos de Máquinas Elétricas. Rio de Janeiro: LTC, 1994.

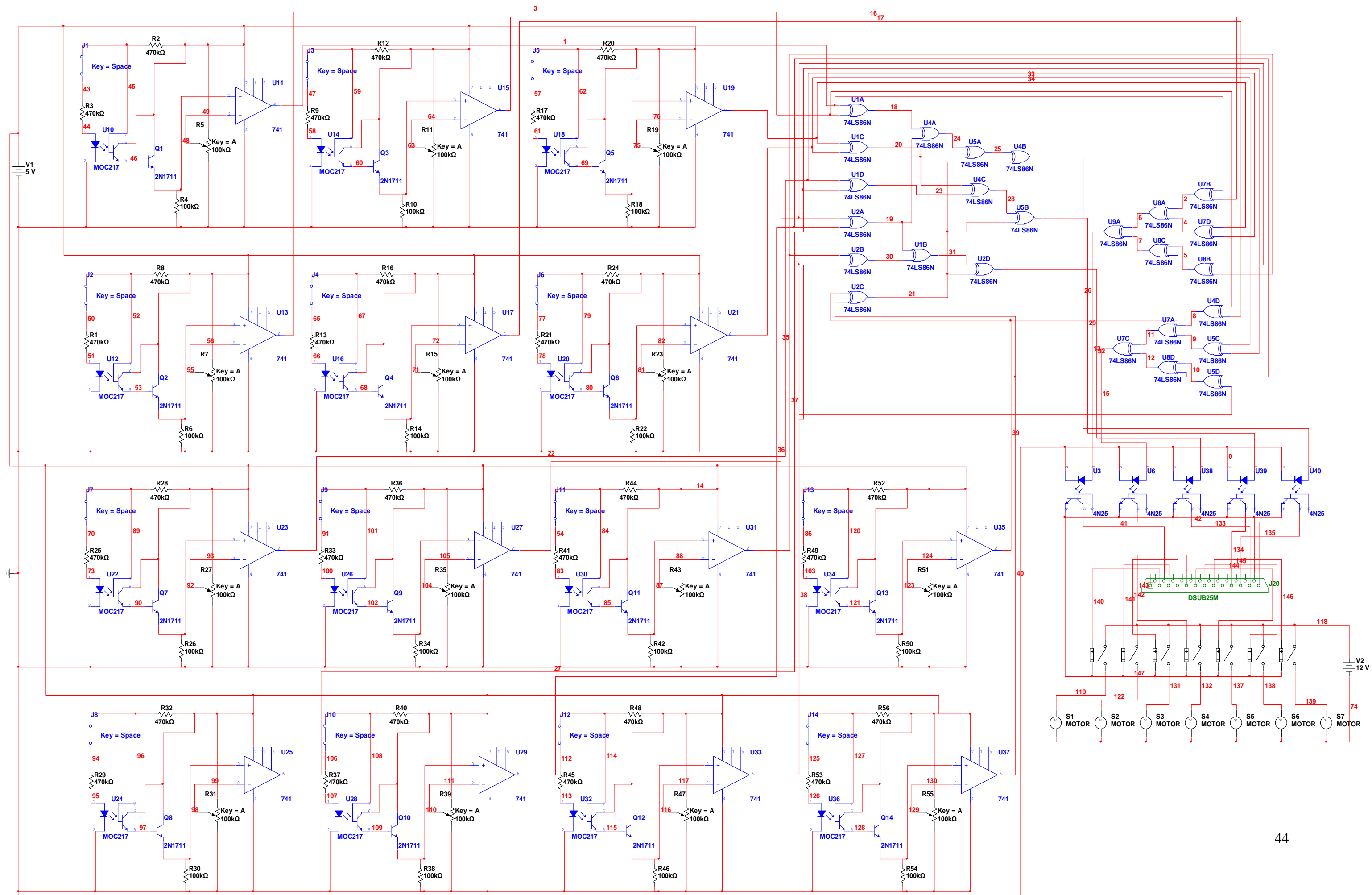
[24] - TORRES, Gabriel, Hardware Curso Completo, 4ª Edição: Axcel Books, 2001.

[25] – WOLF, Oswald e outros, Algebra Booleana, México: Centro Regional de Ajuda Técnica, 1973

Apêndices

Apêndice A – Esquema do Circuito Elétrico da Maquete

O esquema elétrico a seguir apresenta o circuito elétrico completo da estrutura projetada. O arquivo foi desenvolvido no software Multisim Power Pro Edition, versão 10.0.144, da National Instruments.



Apêndices

Apêndice B – Programa Desenvolvido na Linguagem Java

A seguir é apresentado o código da classe *ConfigSistema.java* do programa desenvolvido em linguagem *Java*.

```
/**  
 * Autor Therence Barros  
 */  
  
package br.com.util;  
  
public class ConfigSistema {  
  
    public static final int[] SAIDA_ESTEIRAS = {1,2,4,8,16,32,64,128};  
    //Valores de saída do sistema que ligam as esteiras  
  
    public static final int[][] ENTRADA_ESTEIRAS = {{ 48, 40 },{ 112, 104 },{  
176, 168},{ 240, 232},  
        { 16, 8 },{ 80, 72 },{ 144, 136},{ 192, 216},{ 96, 1}}; //Valores de  
    entrada do sistema. Cada valor represente um sensor IR  
  
    public static final int PORTA_LPT1 = 0x378; //Valor Hexadecimal da  
    porta LPT1  
}
```

Apêndices

Apêndice C – Programa Desenvolvido na Linguagem Java

A seguir é apresentado o código da classe *Tela.java* do programa desenvolvido em linguagem *Java*.

```
package br.com.inicio;

import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;

import javax.swing.AbstractButton;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

import br.com.threads.ThreadLoop;
import br.com.util.ConfigSistema;

public class Tela extends JPanel implements ActionListener {
```



```

/**
 * Autor Therence Barros
 */

private static final long serialVersionUID = 1L;

//Cria as variáveis que irá exibir na tela o conteúdo de cada esteira
private static JLabel jLabel_1 = null;
private static JLabel jLabel_2 = null;
private static JLabel jLabel_3 = null;
private static JLabel jLabel_4 = null;
private static JLabel jLabel_5 = null;
private static JLabel jLabel_6 = null;
private static JLabel jLabel_7 = null;

//Cria os botoes para inicio e fim da execução do programa.
private static JButton jbnInicio, jbnFim;

//Cria o objeto ThreadLoop
private static ThreadLoop threadLoop;

public Tela(){

    //Cria o Layout da tele

    GridLayout experimentLayout = new GridLayout(8,0);

    this.setLayout(experimentLayout);

    add(getJLabel_1());

```

```

        add(getJLabel_2());
        add(getJLabel_3());
        add(getJLabel_4());
        add(getJLabel_5());
        add(getJLabel_6());
        add(getJLabel_7());

        JPanel jPanelBotao = new JPanel(new GridLayout(0,2));
        jPanelBotao.add(getJbnInicio());
        jPanelBotao.add(getJbnFim());

        add(jPanelBotao);

    }

    public void actionPerformed(ActionEvent e) {

        //cria a ação dos botões.

        if ("inicia".equals(e.getActionCommand())) {

            threadLoop = new
ThreadLoop(this,"thread",ConfigSistema.PORTA_LPT1);

```

```

        threadLoop.start();

        jbnInicio.setVisible(false);

    } else if("fim".equals(e.getActionCommand())){

        threadLoop.stop();
        threadLoop.resume();
        jbnInicio.setVisible(true);

    }

}

//Constroi todos os objetos da tela
public JButton getJbnInicio(){

    if(jbnInicio == null){

        jbnInicio = new JButton("Inicia Processo");

        jbnInicio.setVerticalTextPosition(AbstractButton.CENTER);
        jbnInicio.setHorizontalTextPosition(AbstractButton.LEADING);
        jbnInicio.setMnemonic(KeyEvent.VK_I);
    }
}

```

```

        // Alt-I clicks the button

        jbnInicio.setActionCommand("inicia");
        jbnInicio.setToolTipText("Inicia Processo");

        jbnInicio.addActionListener(this);

    }

    return jbnInicio;
}

public JButton getJbnFim() {

    if(jbnFim == null){

        jbnFim = new JButton("Finaliza processo");
        jbnFim.setVerticalTextPosition(AbstractButton.BOTTOM);
        jbnFim.setHorizontalTextPosition(AbstractButton.CENTER);
        jbnFim.setMnemonic(KeyEvent.VK_F);

        // Alt-F clicks the button

        jbnFim.setActionCommand("fim");
        jbnFim.setToolTipText("Finaliza processo");

        jbnFim.addActionListener(this);
    }
}

```

```

    }

    return jbnFim;
}

public static JLabel getJLabel_1() {
    if(jLabel_1 == null){
        jLabel_1 = new JLabel();
        jLabel_1.setSize(30, 50);
        jLabel_1.setText("Esteita 1");
    }
    return jLabel_1;
}

public static JLabel getJLabel_2() {
    if(jLabel_2 == null){
        jLabel_2 = new JLabel();
        jLabel_2.setSize(30, 50);
        jLabel_2.setText("Esteita 2");
    }
    return jLabel_2;
}

```

```
public static void setJLabel_2(JLabel label_2) {  
    jLabel_2 = label_2;  
}
```

```
public static JLabel getJLabel_3() {  
    if(jLabel_3 == null){  
        jLabel_3 = new JLabel();  
        jLabel_3.setSize(30, 50);  
        jLabel_3.setText("Esteita 3");  
    }  
    return jLabel_3;  
}
```

```
public static void setJLabel_3(JLabel label_3) {  
    jLabel_3 = label_3;  
}
```

```
public static JLabel getJLabel_4() {  
    if(jLabel_4 == null){  
        jLabel_4 = new JLabel();  
        jLabel_4.setSize(30, 50);  
        jLabel_4.setText("Esteita 4");  
    }  
}
```

```

    }

    return jLabel_4;
}

public static void setJLabel_4(JLabel label_4) {

    jLabel_4 = label_4;

}

public static JLabel getJLabel_5() {

    if(jLabel_5 == null){

        jLabel_5 = new JLabel();

        jLabel_5.setSize(30, 50);

        jLabel_5.setText("Esteita 5");

    }

    return jLabel_5;

}

public static void setJLabel_5(JLabel label_5) {

    jLabel_5 = label_5;

}

public static JLabel getJLabel_6() {

```

```

        if(jLabel_6 == null){

            jLabel_6 = new JLabel();

            jLabel_6.setSize(30, 50);

            jLabel_6.setText("Esteita 6");

        }

        return jLabel_6;
    }

    public static void setJLabel_6(JLabel label_6) {

        jLabel_6 = label_6;

    }

    public static JLabel getJLabel_7() {

        if(jLabel_7 == null){

            jLabel_7 = new JLabel();

            jLabel_7.setSize(30, 50);

            jLabel_7.setText("Esteita 7");

        }

        return jLabel_7;

    }

    public static void setJLabel_7(JLabel label_7) {

```



```

        jLabel_7 = label_7;
    }

    public void setJbnFim(JButton jbnFim) {
        Tela.jbnFim = jbnFim;
    }

    public void setJbnInicio(JButton jButton){
        jbnInicio = jButton;
    }

    public void setJLabel_1(JLabel label_1) {
        jLabel_1 = label_1;
    }

    private static void createGUI() {
        JFrame.setDefaultLookAndFeelDecorated(true);
        JFrame frame = new JFrame("Controle de Esteiras");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Tela Tela = new Tela();
        Tela.setOpaque(true);
        frame.setContentPane(Tela);
        frame.pack();
        frame.setVisible(true);
    }

```

```
}

public static void main(String[] args) {
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createGUI();
        }
    });
}

}
```

Apêndices

Apêndice D – Programa Desenvolvido na Linguagem Java

A seguir é apresentado o código da classe *ThreadLoop.java* do programa desenvolvido em linguagem *Java*.

```
/**  
 * Autor Therence Barros  
 */  
  
package br.com.threads;  
  
import parport.ParallelPort;  
import br.com.inicio.Tela;  
import br.com.util.ConfigSistema;  
  
public class ThreadLoop extends Thread {  
    //Cria os Objetos da Classe  
    private Object instancia;  
    private int saida;  
    private ParallelPort parport;  
    private boolean[] estadoEsteira = { false, false, false, false, false,  
                                         false, false, false };
```

```
private int[] esteira = ConfigSistema.SAIDA_ESTEIRAS;
```

```
//Construtor da Classe
```

```
public ThreadLoop(Object instancia, String str, int porta) {
```

```
    super(str);
```

```
    this.instancia = instancia;
```

```
    switch (porta) {
```

```
        case ConfigSistema.PORTA_LPT1:
```

```
            if (this.getParport() == null) {
```

```
                this.setParport(new
```

```
ParallelPort(ConfigSistema.PORTA_LPT1));
```

```
            }
```

```
            break;
```

```
        default:
```

```
            System.out.println("A porta " + porta + " não está cadastrada !");
```

```
            System.out.println("Nenhuma porta foi aberta !");
```

```
            break;
```

```
    }
```

```
}
```

```
//Start da Thread
```

```
public void run() {
```

```

boolean continua = true;

Tela tela = (Tela) this.instancia;

boolean bloqueiaControle = false;

int valorPorta = 0;

int[] objetosEsteira = new int[8];

int[] totalObjetos = new int[8];

//cria um loop infinito que fica lendo a porta LPT1
while (continua) {

    //Le a porta LPT1

    valorPorta = parport.read();


    //verifica se a porta está bloqueada para leitura. Evita que o
mesmo valor seja lido 2 vezes.

    if (!bloqueiaControle) {

        //For que verifica se o valor que foi lido na porta é valido

        for (int esteira = 0; esteira <
ConfigSistema.ENTRADA_ESTEIRAS.length; esteira++) {

            for (int sensor = 0; sensor <
ConfigSistema.ENTRADA_ESTEIRAS[esteira].length; sensor++) {

                if (valorPorta ==
ConfigSistema.ENTRADA_ESTEIRAS[esteira][sensor]) {

                    //Verifica se o sensor que está
enviando o sinal é de entrada ou de saída e regista no contador de cada esteira

                    if (sensor == 0) {

```

```

        objetosEsteira[esteira]++;
        totalObjetos[esteira]++;
    } else if (sensor == 1) {
        objetosEsteira[esteira]--;
    } else {
        //System.out.println("Sensor
não Cadastrado !");

    }
    bloqueiaControle = true;
}
}
}

//processa a saída da esteira
saida = EstadoEsteiras(objetosEsteira, saida);

//Escreve na porta LPT1 a saída com o estado de todas
as esteiras

parport.write(saida);
}

//Libera o processo para continuar lendo a porta LPT1
if (bloqueiaControle) {
    bloqueiaControle = LiberaControle(valorPorta);
}

```

```

        //Escreve na tela de saida para o usuário a quantidade de
objetos em cada esteira

        for (int i = 0; i < objetosEsteira.length; i++) {

            //System.out.println("Esteira " + (i + 1) + " possui " +
objetosEsteira[i] + " objetos total "+totalObjetos[i]);

            if (i == 0) {

                tela.getJLabel_1().setText("Esteira 1 possui " +
objetosEsteira[i] + " objetos total "+totalObjetos[i]);

                tela.getJLabel_1().repaint();

            }

            if (i == 1) {

                tela.getJLabel_2().setText("Esteira 2 possui " +
objetosEsteira[i] + " objetos total "+totalObjetos[i]);

                tela.getJLabel_2().repaint();

            }

            if (i == 2) {

                tela.getJLabel_3().setText("Esteira 3 possui " +
objetosEsteira[i] + " objetos total "+totalObjetos[i]);

                tela.getJLabel_3().repaint();

            }

```

```
        if (i == 3) {  
            tela.getJLabel_4().setText("Esteira 4 possui " +  
objetosEsteira[i] + " objetos total "+totalObjetos[i]);  
            tela.getJLabel_4().repaint();  
        }  
  
        if (i == 4) {  
            tela.getJLabel_5().setText("Esteira 5 possui " +  
objetosEsteira[i] + " objetos total "+totalObjetos[i]);  
            tela.getJLabel_5().repaint();  
        }  
  
        if (i == 5) {  
            tela.getJLabel_6().setText("Esteira 6 possui " +  
objetosEsteira[i] + " objetos total "+totalObjetos[i]);  
            tela.getJLabel_6().repaint();  
        }  
  
        if (i == 6) {  
            tela.getJLabel_7().setText("Esteira 7 possui " +  
objetosEsteira[i] + " objetos total "+totalObjetos[i]);  
            tela.getJLabel_7().repaint();  
        }
```



```

    }

    try {

        //System.out.println("parada do loop");

        Thread.sleep(100);

        //System.out.println("inicio do loop");

    } catch (Exception e) {

        e.printStackTrace();

    }

}

}

/*
 * Verifica se a o valor lido na porta mudou, desta forma
 * caso o objeto seja grande não é lido 2 vezes o mesmo valor na porta.
 */

private boolean LiberaControle(int valorPorta) {

    for (int esteira = 0; esteira <
ConfigSistema.ENTRADA_ESTEIRAS.length; esteira++) {

        for (int sensor = 0; sensor <
ConfigSistema.ENTRADA_ESTEIRAS[esteira].length; sensor++) {

```

```

        if (valorPorta ==
ConfigSistema.ENTRADA_ESTEIRAS[esteira][sensor]) {

            //System.out.println("Sensor " + (esteira + 1) +
(sensor + 1) + " - " + ConfigSistema.ENTRADA_ESTEIRAS[esteira][sensor]);

            return true;

        }

    }

}

return false;

}

```

//Desliga todas as esteiras

```

public void RecetPortas() {

    getParport().write(0);

    estadoEsteira = new boolean[] { false, false, false, false, false,
false, false, false };

}

```

```

public void StartTodasPortas() {

    getParport().write(255);

}

```

//testa a saída de todas as esteiras

```
public void TestaSaidas() {  
    for (int i = 0; i < 9; i++) {  
        int valor = 0;  
        if (i == 0) {  
            valor = 1;  
        } else if (i == 1) {  
            valor = 2;  
        } else if (i == 2) {  
            valor = 4;  
        } else if (i == 3) {  
            valor = 8;  
        } else if (i == 4) {  
            valor = 16;  
        } else if (i == 5) {  
            valor = 32;  
        } else if (i == 6) {  
            valor = 64;  
        } else if (i == 7) {  
            valor = 128;  
        } else if (i == 8) {  
            valor = 255;  
        }  
        getParport().write(valor);  
        try {
```

```

        Thread.sleep(500);
    } catch (Exception e) {
    }
}

try {
    Thread.sleep(1000);
} catch (Exception e) {
}

getParport().write(0);

// porta.write(255);

//System.out.println("Valor na porta de entrada " + getParport().read());
}

/*
 * Processa a saída verificando a quantidade de objetos sobre a esteira
 * e mudando o estado da esteira para ligado ou desligado
 */

public int EstadoEsteiras(int[] objetosEsteira, int ultimaSaida) {

    int retorno = ultimaSaida;

    for (int i = 0; i < objetosEsteira.length; i++) {

        //Liga Esteira

```

```

        if (!estadoEsteira[i] && objetosEsteira[i] == 1) {

            retorno += esteira[i];

            estadoEsteira[i] = true;

        }

        //Desliga Esteira

        if (estadoEsteira[i] && objetosEsteira[i] == 0) {

            retorno -= esteira[i];

            estadoEsteira[i] = false;

        }

        if (retorno < 0) {

            retorno = 0;

        }

    }

    return retorno;

}

public ParallelPort getParport() {

    return parport;

}

public void setParport(ParallelPort parport) {

    this.parport = parport;

}

}

```